

機器學習期末報告

Predict Future Sales

資工系四年 A 班 學號 1061443

資工系四年 A 班 學號 1061416

資工系四年 C 班 學號 1063337

一、組員名單與分工(請標註工作佔比)

- 請詳細列出工作項目，以及在專題的工作佔比

學號	工作內容	專題佔 比
1061443	Survey paper、LSTM 模型實作、製作報告、報告主講、demo 影片	40%
1061416	Survey paper、XGBoost 模型實作	30%
1063337	Survey paper、Random Forest 模型實作	30%

二、開發動機與目標

做銷售量預測，可以對店家有正向的幫助，店家透過預測可以知道有些時段銷售量是高是低，可以根據這個預測去考量進貨的量，減少成本。消費者端也可以透過銷售量預測，或許銷售量低的時候，價格也會促銷，撿到便宜！

我們希望我們這個比賽可以達到比賽前 15%

三、參考文獻探討

-此題目是否有相關的 paper 也是這個題目

Predicting Future Sales of Retail Products using Machine Learning [2020-08-18]

-說明文獻上怎麼做，有什麼缺點

這篇論文也是參與相同的比賽，他們的缺點我覺得在於資料前處理還可以做得更好，shop_id、item_category_id 這兩個欄位他們並沒有去多做處理，我們自己就有多做這方面的處理，再來就是論文中的 lag feature 取 1-3 months 跟 12 months，我覺得可以再取 6 months，以半年來做數據統計也是有意義的。

-用一個表格說明這些文獻與你的方法的差異

論文的方法	我們的方法
shop_id 、 item_category_id 未做處理	利用 shop_id 、 item_category_id 取出 city_code 、 type_code 、 subtype_code
Lag_feature 取 1-3 months 、 12 months	Lag_feature 取 1-3 months 、 6 months 、 12 months

四、 解決方案介紹

- 提出的方法架構

我們使用了三種方法：

1. LSTM
2. XGBoost
3. RandomForest

- 蒐集的資料

使用 Kaggle 競賽提供的資料：

檔案說明：

- **sales_train.csv** - the training set. Daily historical data from January 2013 to October 2015.
- **test.csv** - the test set. You need to forecast the sales for these shops and products for November 2015.
- **sample_submission.csv** - a sample submission file in the correct format.
- **items.csv** - supplemental information about the items/products.
- **item_categories.csv** - supplemental information about the items categories.
- **shops.csv** - supplemental information about the shops.

欄位說明：

- **ID** - an Id that represents a (Shop, Item) tuple within the test set
- **shop_id** - unique identifier of a shop

- **item_id** - unique identifier of a product
- **item_category_id** - unique identifier of item category
- **item_cnt_day** - number of products sold. You are predicting a monthly amount of this measure
- **item_price** - current price of an item
- **date** - date in format dd/mm/yyyy
- **date_block_num** - a consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33
- **item_name** - name of item
- **shop_name** - name of shop
- **item_category_name** - name of item category

- 資料前處理

1. 觀察資料數值分佈，去除離峰值。
2. 觀察資料欄位。若發現 NA，則根據商店、商品以及日銷售量的欄位取平均值來填補。
3. 將店名相同但商店 id 不同的資料統一。
4. 根據店家名稱，分析商店所在城市。
5. 根據商品項目，分析主要類型以及次要類型。
6. 對城市、商品主要類型和次要類型做 label encoding。
7. 根據商店和商品資料計算商品的月銷售量，並將月銷售量小於 0 的欄位補 0，大於 20 的欄位補 20。(比賽有特別說明月銷售量只會介於 0 到 20 之間)
8. 透過 mean encoding 的方法增加資料欄位。針對日銷售量、商品 id、商店 id、商品項目 id、商品主要類型、商品次要類型的欄位以不同方

式組合做 mean encoding。

9. 計算商品日銷售額和同商品平均銷售額之間的差異。
10. 計算商店日銷售額和同商店平均銷售額之間的差異。
11. 計算每間商店、每樣商品和前一次售出日日銷售量之間的差異。
12. 計算每樣商品和前一次售出日日銷售量之間的差異。
13. 計算每間商店、每樣商品和售出最少日日銷售量之間的差異。
14. 計算每樣商品和售出最少日日銷售量之間的差異。
15. 增加月份以及當月天數的欄位。

總體來說，資料前處理部分有去除離峰值、填補空缺值、針對有字串的欄位做 label encoding、透過 mean encoding 的方法增加新欄位、以不同方法計算日銷售額的差異以及增加日期天數的欄位。

除此之外，將 34 個月的資料內容切割成前 32 個月訓練集，第 33 個月驗證集和第 34 個月測試集。由於前 3 個月在產生新欄位時，欄位內容無可追溯性，所以刪除前 3 個月的資料。

- 模型說明

LSTM

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 64)	16896
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65
Total params: 16,961		
Trainable params: 16,961		
Non-trainable params: 0		

- 1 LSTM layer
- 1 Dropout layer
- 1 Dense layer
- Loss: root mean squared error
- Optimizer: adam
- Batch_size: 4096
- Epochs: 10

XGBoost

- max_depth=12
- n_estimators=300
- min_child_weight=162
- colsample_bytree=0.6
- subsample=0.8
- eta=0.008
- seed=42



Randomforest

- max_depth=8
- n_estimators=70
- random_state=0
- n_jobs=-1

五、模型預測結果

- 參賽結果

於 12726 名對手中排名第 1576 名，成績為 0.89647。

1576	peter0512lee		0.89647	26	3d
Your Best Entry 					
Your submission scored 0.90092, which is not an improvement of your best score. Keep trying!					

- 比較多個模型的預測效果

	Kaggle Score		
LSTM	1.01950		
XGBoost	0.89647	1.02610 (w/o lag feature)	0.92784 (paper)
RandomForest	0.94685	0.93215 (w/o lag feature)	

從表格可以得知，XGBoost 的效果最好，以及有沒有做 lag feature 其實對分數是有差距的，有做 lag feature 不管是在哪個模型，分數都有顯著的提升！

六、 開發最耗時的部份與原因

- 遇到的問題

這比賽的資料是俄文，像是 shop_name、item_category_name，這些欄位其實都是俄文，我們又要取出店名中城市的部分，所以我們就瘋狂翻譯 X D

- 未來可改善的地方

LSTM 我覺得很有機會做得更高，我們光是沒做什麼前處理就有 1.02 score，我覺得未來前處理做多一點很有機會超越 XGBoost

七、 小組互動照片

- 合照、討論專題時的照片



八、專案開發心得

1061443 - 專題開發心得

這應該算是我第一次正式參加完整個 kaggle 比賽，成績方面我其實算是蠻滿意的，有達到我們當初設下的目標，前 15%，可喜可賀。

透過這次的比賽其實學到很多工具跟資料前處理的方法，工具的部分我們有使用 optuna 這個框架去做參數優化，如果我今天沒做這個專案我還真的不知道 X D，這個框架真的很好用，大家可以去用用看！再來就是資料前處理，lag feature 也是我第一次接觸，原來 lag feature 在時間序列分析這方面的問題是這麼有用的，不管是什麼模型，有做 lag feature 就是硬加個 10%，真的很厲害，也讓我體認到模型優化其實不是最重要的，資料前處理也是很重要的一環。最後感謝我的隊友互相分工，感謝各位！

1061416 - 專題開發心得

這是我第一次參加 kaggle 上的比賽。由於 kaggle 平台上有些其他人的開源程式碼可以參考，所以我可以透過平台上的開源程式碼，學習到時序性資料的許多處理方式，也可以認識到哪種類型的題目，大家都會選用甚麼樣的模型。在尋找相關資料時，我看到很多人都有使用 lag feature 和 mean encoding 的方法來增加資料量，於是我和我的組員也決定實做這個方法。整體的資料前處理，我們採用了平常比較常聽到的 label encoding 的方法，並去除離峰值、填補缺失值，再加上 mean encoding 的方法增加新欄位。我們也有比較過有使用 mean encoding 方法和沒有使用 mean encoding 方法的差異，發現有使用 mean encoding 的方法效果真的會好很多。同時，在查詢資料的過程中，我們也有找到 optuna 這個參數調整的架構。透過 xgboost 加上 optuna，我們可以輕鬆地找出最佳的參數組合，大幅省去我們嘗試不同參數的時間。整體來說，整個比賽過程獲益良多。為了提升準確度，嘗試多種不同的前處理方式，也更加了解模型裡面參數的意思。

1063337 - 專題開發心得

資料收集過程中首次接觸到 lag feature 與 mean encoding，經過比較後發現透過這些 feature 可以讓模型的準確度大幅提升，這時候才發現自己對於資料前處理技術的視野仍過於狹窄。透過組員介紹而得知 Optuna，比較可惜的是我所負責的 RandomForest Regression 無法透過 GPU 訓練，難以藉由 Optuna 來自動優化參數。在 kaggle 中大致瀏覽過就會發現，在參考其他高手分享的 EDA 的時候，也要把下方的 comment 看完，在 comment 中能看到他人的提問，當我看完原作者的解答後可以快速瞭解原始碼其中的細節。課程中所學的模型大部份都是第一次使用，透過期末專題將不同模型、不同前處理做交叉測試與比較，讓我更加瞭解機器學習這門學問的意義，而 lag feature、mean encoding 與 Optuna 是本次專案中最大的收穫。